FINDING A REPRESENTATIVE DAY FOR SIMULATION ANALYSES

A Thesis
Presented to
The Academic Faculty

by

Jebulan Ryan Watson

In Partial Fulfillment
of the Requirements for the Degree
Master's of Science in the
School of Aerospace Engineering

Georgia Institute of Technology
December 2009

FINDING A REPRESENTATIVE DAY FOR SIMULATION ANALYSES

Approved by:

Dr. John-Paul Clarke, Advisor
School of Aerospace Engineering
Georgia Institute of Technology

Dr. Eric Feron
School of Aerospace Engineering
Georgia Institute of Technology

Dr. Ellis Johnson
School of Industrial and Systems Engineering
Georgia Institute of Technology

Date Approved: November 4th, 2009

This thesis work is dedicated to my family, Eugene, Madonna, and Kelsey Watson, whose love and support helped me get where I am today.

ACKNOWLEDGEMENTS

There are many people that I would like to thank for their help. The first person I would like to thank is my Research Advisor, Dr. John-Paul Clarke. It was Dr. Clarke who hired me as an undergraduate and kept me as a graduate student. Dr. Clarke pointed me in the direction of this thesis topic, for which I must give him thanks. Terran Melconian also helped me when I was working on this thesis. He introduced me to the wonders of the language of Python, the foundation of my work. Without his help, I would not be able to write code or analyze data the way I can today. He also put up with all of the questions and problems that arose while I was working on this thesis.

Next I want to thank my Advisory Committee, which consisted of the aforementioned Dr. Clarke, Dr. Eric Feron, and Dr. Ellis Johnson. They have all helped me grow, whether it was through having them as a professor, or talking to them at their office or at meetings. With their suggestions at my thesis proposal, I was able to refine my work.

Finally I would like to thank everyone else who provided me with moral support. Jim Brooks, a research engineer in the Air Transportation Laboratory (ATL), taught me innumerable amounts of information from all of his experiences. I don't think I can fit the names of everyone from the ATL into this, but every one of them spent time with me over the course of my four years working there and became friends with me. Also, I have to thank my family and friends, who supported me as well. This thesis may have my name on it, but every one of these people helped in their own ways to make this thesis possible.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Page

LIST OF SYMBOLS AND ABBREVIATIONS

SF_i	Score Factor dependent on quantity *i*
μ_{day}, μ_{year}	Mean
σ_{day}, σ_{year}	Standard Deviation
X_{day}, X_{year}	Distribution of quantity
n_{day}, n_{year}	Number of cancellations
C_i	Weighting Coefficient of quantity *i*
ACES	Airspace Concept Evaluation System
ARTCC	Air Route Traffic Control Center
ATL	Air Transportation Laboratory
ATM	Air Traffic Management
BTS OTP	Bureau of Transportation Statistics Airline On-time Performance
CSV	Comma-Separated Values
GUI	Graphic User Interface
MEANS	MIT Extensible Air Network Simulation
NAS	National Airspace System
NASPAC	National Airspace Performance Capability
NOAA	National Oceanic and Atmospheric Administration
OUT	Output (file type)
RDF	Representative Day Finder
SIMMOD	Simulation Model
TAAM	Total Airspace and Airport Modeler
TRACON	Terminal Radar Approach Control

SUMMARY

Many models exist in the aerospace industry that attempt to replicate the National Airspace System (NAS). The complexity of the NAS makes it a system that can be modeled in a variety of ways. While some NAS models are very detailed and take many factors into account, runtime of these simulations can be on the magnitude of hours (to simulate a single day). Other models forgo details in order to decrease the runtime of their simulation. Most models are capable of simulating a 24 hour period in the NAS. An analysis of an entire year would mean running the simulation for every day in the year, which would result in a long run time.

The following thesis work presents a tool that is capable of giving the user a day that can be used in a simulation and will produce results similar to simulating the entire year. Taking in parameters chosen by the user, the tool outputs a single day, multiple days, or a composite day (based on percentages of days). Statistical methods were then used to compare each day to the overall year. On top of finding a single representative day, the ability to find a composite day was added. After implementing a brute force search technique to find the composite day, the long runtime was deemed inconvenient for the user. To solve this problem, a heuristic search method was created that would search the solution space in a short time and still output a composite day that represented the year. With a short runtime, the user would be able to run the program multiple times. Once the heuristic method was implemented, it was found that it performed well enough to make it an option for the user to choose.

The final version of this tool was used to find a representative day and the result was used in comparison with output data from a NAS simulation model. Because the tool found the representative day based on historical data, it could be used to validate the effectiveness of the simulation model. The following thesis will go into detail about how this tool, the Representative Day Finder, was created.

CHAPTER 1

INTRODUCTION

Modeling and simulating the National Airspace System (NAS) has been a task in which people in the aerospace industry have put their time and effort. The NAS itself is a complex system which takes into account many factors, causing a tradeoff between runtime and detail to arise. If one wants to accurately model the NAS, a model must be used that is able to represent the NAS as close as possible. By increasing the detail of the model, the run time of the model will increase, due to the amount of information that must be accounted for and processed. On the other side of the spectrum, a less detailed model will result in a faster run time. Depending on the detail of the model, the run time of an analysis can be short or long.

For example, the Airspace Concept Evaluation System (ACES) is a simulation model of the NAS that keeps track of flights and their trajectories throughout the NAS for an entire day. Depending on demand and capacity (as well as other factors), the simulation time for an ACES run can take six hours or longer [1]. If an analysis for an entire year was to be done with ACES (a run for every day of the year), the analysis would take roughly three months to complete. The MIT Extensible Air Network Simulation (MEANS), on the other hand, does not keep track of en route details to the extent ACES does. MEANS was made for fast simulation times, so an average MEANS run can take about five minutes [2]. An analysis for an entire year using MEANS would take about a day and a half. Although the overall time is shorter, less information can be outputted, showing there is a tradeoff between simulation time and detail.

Rather than running every day of an entire year for an analysis, it could be possible to find a day that represents the entire year. When a project wants to look at future growth, a baseline scenario is chosen [16]. This baseline is then grown into future scenarios to analyze. This means the analysis is now looking at a baseline scenario and a few out-year scenarios. On top of that, there may be deviations from the baseline that will be looked into, as well as the growth of those cases. This easily makes the scenarios to analyze grow out of hand. Doing this for 365 days and then taking runtime into account, a large amount of time will be spent running the scenarios. Finding a representative day reduces this case of many scenarios to only one.

The argument that it is not possible for a single day to represent an entire year now arises [3]. It is true that a single day can not perfectly represent an entire year when taking into account every measurable quantity. However, if certain aspects of the year were focused on, then a single day could possibly represent a year. As a small example, if the quantities that were being measured were delay, cancellations, and weather, then it would be improbable that a day exists that can represent all of these quantities for the entire year. Now, if we focus only on cancellations and disregard the other two quantities, then it is more likely that we can find a day that represents the year.

This thesis has created a tool, the Representative Day Finder (RDF), which allows the user to choose quantities of interest and in return, receive a day that best represents the year with respect to their chosen quantities. With the ability to find a single day (or set of days), the run time for an analysis of the whole year should be drastically reduced. Another benefit from only running one day is that errors in the simulation run can be made with less of an impact on the analysis. If running an entire year takes three months and an error is found after the runs have completed, then another three months will be spent running the simulation again. Only running one day means errors found will not be as costly as for running every day. The rest of this paper will go into detail on how the RDF was created, how it works, and its success in finding a representative day.

CHAPTER 2

BACKGROUND

The following chapter will provide background information regarding subjects related to this thesis. Because the RDF will reduce run time of simulations, information about some currently used simulation tools is provided. The RDF also takes historical data into account when finding a representative day. This historical data comes in the form of BTS OTP data, about which more information follows.

2.1 SIMULATION MODELS

2.1.1 MEANS

The MIT Extensible Air Network Simulation (MEANS) is a simulation tool capable of modeling the NAS [2]. MEANS is able to track capacity, delay, and passengers within the system for a single day. MEANS is made up of seven modules that perform different tasks in modeling the system. Additional modules can be created based on the needs of the user.

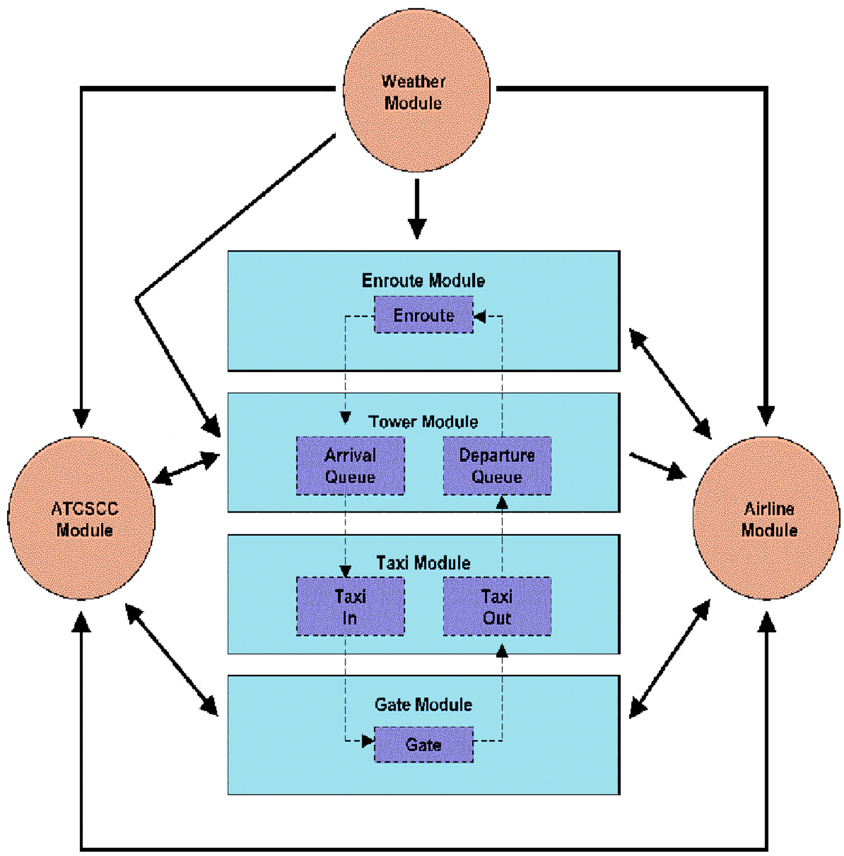

Figure 2.1: Module breakdown of how MEANS models the NAS [18].

MEANS can simulate specific days in history as well as user inputted days. From historical data, schedule and weather files can be generated and used for inputs into a MEANS simulation. A single simulation can take up to five minutes (larger schedules take more time) to run, so running an entire year would take about thirty hours. Arrival and departure delay is easily obtained from MEANS output data.

The output data from MEANS consists of information about each scheduled flight from the input schedule files. Basic information about the flight, such as flight number and departure/arrival airports are given. Also, time stamps of the aircraft location in the terminal area are provided. For example, the time the aircraft begins taxiing out, the time the aircraft arrives at the departure queue, and the time the aircraft lifts off can be found in the output data (also for the arrival process). Delay can be calculated from the data in the MEANS output files.

2.1.2 Other Models

The Airspace Concept Evaluation System (ACES) is another simulation tool that models the NAS [1]. ACES simulates flights from pushback arrival at the gate, as well as operations while the flight is en route. Flight trajectories and weather are also included in the simulation. A major use for ACES is evaluating the impact of new technologies on the NAS through simulation.

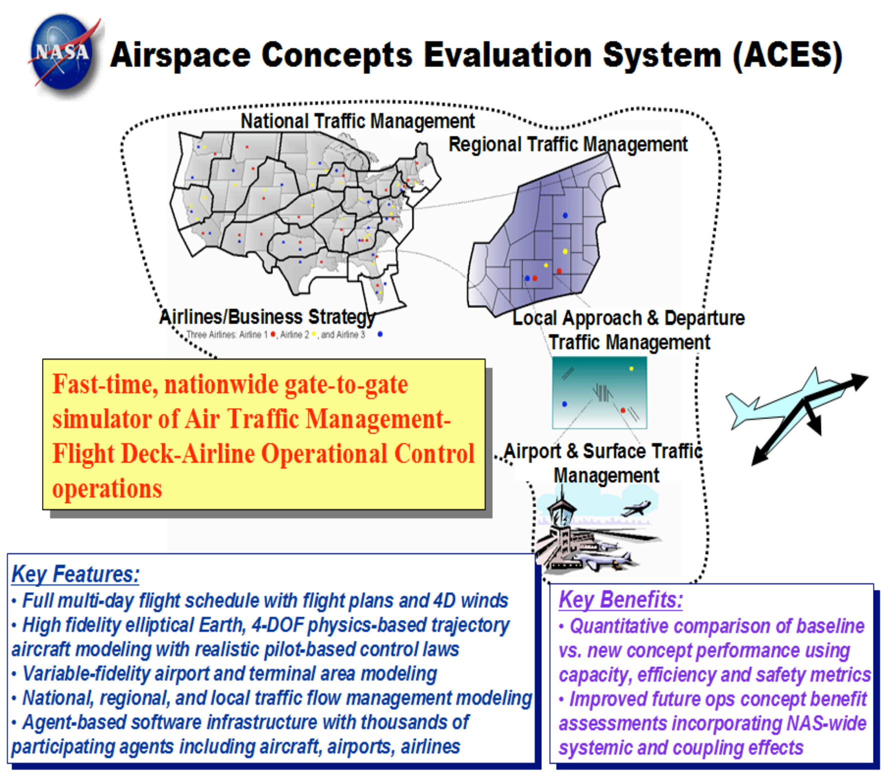

Figure 2.2: Description of ACES capabilities and benefits [18].

ACES is divided into Air Traffic Management (ATM) modeling and non-ATM modeling. The ATM modeling includes models of the airport, Terminal Radar Approach Control (TRACON), and Air Route Traffic Control Center (ARTCC), as well as others.

The non-ATM models include traffic demand, flight trajectories, and weather. The most recent version of ACES requires around seven hours for completing a simulation. As with MEANS, run time for ACES is also dependent on demand and capacity.

There are many other models that simulate the NAS. Simulation Model (SIMMOD) is another detailed model that simulates airport and airspace operations [4, 5]. Total Airspace and Airport Modeler (TAAM) is also very detailed, and it can be used in conjunction with SIMMOD [6]. The National Airspace Performance Capability (NASPAC) simulation, while not as detailed as SIMMOD or TAAM, is a module based model that can estimate the impact of certain actions on the NAS [7, 8]. A representative day would be ideal to help reduce the number of simulations that would have to run in an analysis with these models.

2.2 BTS OTP DATA

The Bureau of Transportation Statistics Airline On-time Performance (BTS OTP) data that was used is data from the Trans Stats data library [9]. This data comes from the On-Time Performance data table of the "On-Time" database. This historical data is separated by month for a given year. The super computer used by the Air Transportation Laboratory (ATL) houses BTS OTP data starting from 1987 and ending with 2006 (at the time of this thesis).

The BTS OTP data itself contains 55 different fields of information on scheduled flights throughout the country. The BTS OTP data contains the pertinent information about the flight. For example, basic information such as arrival/departure airports and times are provided. The fields of information that can be found within the data that are of interest are arrival delay, departure delay, weather delay, taxi-out time, and cancellations.

CHAPTER 3

SUPPORTING CONCEPTS

The following sections of this chapter will provide definitions and information on different concepts used in the RDF. The information at times may provide a brief explanation, so further detail can be found through the listed references or the reader's own search method.

3.1 STATISTICAL TERMINOLOGY

In order to find a representative day in a year, each day needed to be compared to the year itself. To do this, statistical information was used to make a comparison. The most common quantity to use for comparison is the mean. The mean is defined as the sum of a sample divided by the size of the sample [10]. By making the sample contain every day of the year, the mean of the year can be found. After finding the mean of the year, each day can be compared to the year's mean in order to see which day is most similar to the year.

Another statistical quantity that provides use for comparison is the standard deviation. The standard deviation is a quantity that measures variability within a sample of data. The sample mean must be known in order to acquire the standard deviation of the sample. To calculate the standard deviation, the difference between the mean and observation squared is calculated, then summed over the entire sample, and finally divided by the number of samples. Knowing how to find the standard deviation can help when comparing two distributions. While the mean and standard deviation are commonly used comparison quantities, they do not do a thorough job of comparing each day to the year. To get a better comparison, the distribution can be used.

A distribution is a property of a sample population. A distribution for the year and each day can be found for quantities of interest. For example, total delay of a flight can be considered. For a single day, there is a distribution of total delay that is found from looking at each flight's total delay. The same can be done for the year. A graphical representation of the total delay distribution can be seen by plotting the total delay values along the x-axis and the frequency of those values along the y-axis. Knowing these distributions allow for another means of comparison between a day and the year.

3.2 OPTIMIZATION TECHNIQUES

The subject of optimization branches into many different subdivisions. The first major split is between linear and nonlinear optimization [12]. Linear optimization deals with the optimization of a linear objective/cost function, while nonlinear optimization deals with a nonlinear objective function. After the scores for each day were found during the development of the Perfect Day Finder, the solution space was found to be nonlinear. Due to this finding, quickly searching for the best composite day (when desired) became a difficult task.

Finding the minimum for the solution space became difficult because the global minimum must be found. The nonlinear solution space contains many minima and maxima, so a global search technique needs to be implemented. To begin, a brute force method can be implemented to find the global minimum to the nonlinear solution space. This method is very simple to code, but a tradeoff occurs with computation time. Every possible solution needs to be computed, making the brute force search method very time consuming. The brute force method may be time consuming, however, the minimal solution will be found.

Because computation time can become an issue, other global search techniques have been created [13]. One popular method used for optimization is the genetic search.

This method involves using binary strings to represent variables, then calculating the solution (fitness) for that string. Operations on the string (Reproduction, Crossover, and Mutation) are performed to created new strings and the fitness of the new string is evaluated. This repeats until there is a convergence in the solution.

Another method used in finding a global minimum is the particle swarm. This method entails starting with a random set of variables (the initial swarm). The points in the swarm will then have their solutions calculated. Afterwards, local search techniques can be implemented around the initial swarm points. This will continue until convergence is achieved. A problem that can occur is that the global minimum is not guaranteed to be found (the initial swarm may not have a point that falls in the local area of the global minimum).

Lastly, the nonlinear solution space can be linearly approximated, if possible. Sequential linear programming uses linear approximations as well as the gradient of the function in order to find the solution. In some cases this works very well, however, sometimes the solution found can be infeasible due to the approximations. For the solution sets that appear using the RDF, a heuristic search technique was used to find a composite day to represent the entire year. Further explanation of this heuristic search technique can be found in Section 4.2.3.

CHAPTER 4

FORMULATION DESCRIPTION

This chapter will describe the methods used in the RDF as well as how the RDF works. This was done to make the explanation (of how the RDF works) as sequential as possible.

4.1 INITIAL WORK

4.1.1 Data Comparison

Before comparing individual days and the entire year, the question of what to compare must be answered. Delay is a very important factor that can be used for comparison. Delay varies from day to day, so there should be a day within a given year that has delay close to the average delay of the year (there will always be at least one day closest to the yearly average). Looking at the different types of delay, arrival delay and departure delay are of importance because they account for most delay. Another parameter to consider is the number of cancellations. Looking at the number of cancellations is an indirect way of taking weather into consideration. Days with a high number of cancellations are probably days with a lot of bad weather (assuming non-weather related catastrophic events did not occur). Weather delay is provided in the BTS OTP data, so it can be taken into account as well. Taxi-out delay, while not explicitly given in the BTS OTP data, can be calculated from data within the BTS OTP data (an explanation of how to calculate taxi-out delay can be found in Chapter 4.1.2).

The base of the RDF is using statistical data to compare individual days to the entire year. Finding and comparing the means and standard deviations of the different factors are the starting point of the comparison. Comparing the distributions of these factors is what will lead to a more significant comparison. A day that has distributions matching the year's distributions will be the ideal candidate for the representative day. The key to comparing two distributions is to find the difference in area of the distributions.

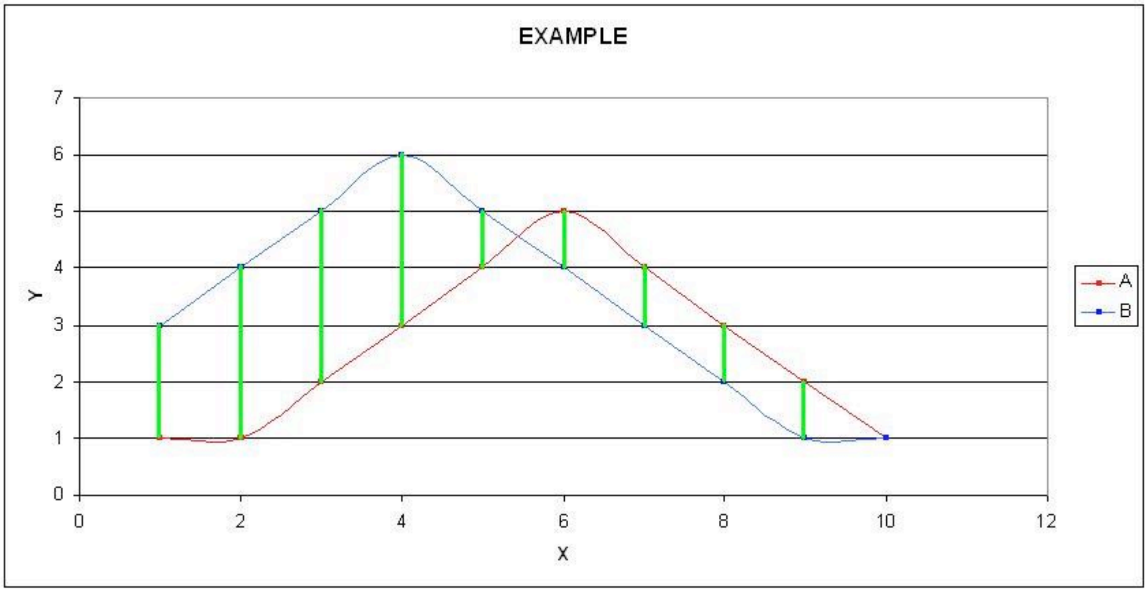

Figure 4.1: Example showing the difference (green lines) between two distributions.

If the distributions are the same, then there is no difference in area. A method to quantify the difference in the distributions is to take the difference between the distributions at each point along the x-axis (green lines in Figure 4.1), square the value, and then sum along the distributions. This results in a numerical value that can be assigned to every day in the year. Days with values near zero are better matches for the year than days with higher values (with zero meaning the two distributions are the same). Using this approach, every day will have an assigned value pertaining to each of the factors of interest (delay, cancellations, etc...).

4.1.2 Taxi-out Delay Calculation

Unlike arrival delay, the taxi-out delay of a flight is not given in the BTS OTP data. Taxi-out time is a quantity that is provided in the BTS OTP data, so a way to derive the taxi-out delay from taxi-out time needs to be formulated. A paper written by Idris et al. contains a procedure for deriving taxi-out delay from taxi-out time [11]. To begin the procedure, the number of aircraft that take off during each flight needed to be found. This was done by using the wheels off time and departure time that can be found in the BTS OTP data. The cutoff amount of aircraft for each airport can be calculated by halving the average number of aircraft during takeoff at an airport. Once the cutoff amount is established, the average taxi-out time for flights falling below the cutoff amount is found. This value is the unimpeded taxi-out time as calculated from Idris' procedure.

Following this procedure, the unimpeded taxi-out time can be calculated for each airport. Taking the given taxi-out time provided by the BTS OTP data and subtracting the unimpeded taxi-out time will result in the taxi-out delay for that flight. Currently this is the only quantity from the BTS OTP data that has needed to be converted into a more useful quantity. By having to do this procedure, the program must loop through the BTS OTP data an extra time, which increases the program's runtime.

4.1.3 Scoring Function

The RDF uses thirteen factors to compare each day to the year. Arrival, departure, weather, and taxi-out delay are four of the five main factors that are used for the scoring function. For each type of delay, the mean, standard deviation, and distribution are factored into the equation. The other main factor used is the number of cancellations that occurred. All thirteen factors are used to make the comparison between the day and the year:

$$SF_{ArrivalDelayMean} = (\mu_{day} - \mu_{year})^2_{ArrivalDelay} \tag{4.1.1}$$

$$SF_{ArrivalDelayStandardDeviation} = (\sigma_{day} - \sigma_{year})^2_{ArrivalDelay} \tag{4.1.2}$$

$$SF_{ArrivalDelayDistribution} = (X_{day} - X_{year})^2_{ArrivalDelay} \tag{4.1.3}$$

$$SF_{TaxioutDelayMean} = (\mu_{day} - \mu_{year})^2_{TaxioutDelay} \tag{4.1.4}$$

$$SF_{TaxioutDelayS\tan dardDeviation} = (\sigma_{day} - \sigma_{year})^2_{TaxioutDelay} \tag{4.1.5}$$

$$SF_{TaxioutDelayDistribution} = (X_{day} - X_{year})^2_{TaxioutDelay} \tag{4.1.6}$$

$$SF_{WeatherDelayMean} = (\mu_{day} - \mu_{year})^2_{WeatherDelay} \tag{4.1.7}$$

$$SF_{WeatherDelayS\tan dardDeviation} = (\sigma_{day} - \sigma_{year})^2_{WeatherDelay} \tag{4.1.8}$$

$$SF_{WeatherDelayDistribution} = (X_{day} - X_{year})^2_{WeatherDelay} \tag{4.1.9}$$

$$SF_{DepartureDelayMean} = (\mu_{day} - \mu_{year})^2_{DepartureDelay} \tag{4.1.10}$$

$$SF_{DepartureDelayS\tan dardDeviation} = (\sigma_{day} - \sigma_{year})^2_{DepartureDelay} \tag{4.1.11}$$

$$SF_{DepartureDelayDistribution} = (X_{day} - X_{year})^2_{DepartureDelay} \tag{4.1.12}$$

$$SF_{Cancellations} = (n_{day} - n_{year})^2 \tag{4.1.13}$$

The comparison for the mean, standard deviation, and number of cancellations consisted of taking the difference between the day and the year and squaring it. For the comparison of the distributions, Section 4.1.1 provides the explanation of the method used.

Once the comparison is made, each day is given a score factor associated with the quantities that were measured. The values for the weighting coefficients, C_i, are inputted by the user and the quantity score factors are calculated within the RDF. The scoring function used is as follows:

$$\min f = \sum_i C_i SF_i \tag{4.1.14}$$

The scoring function will produce a value for every day of the year. Once these values are calculated, the minimum value is found, and its associated day will be the representative day for the year.

4.2 THE REPRESENTATIVE DAY FINDER

The RDF consists of a single executable Python code. The Tkinter module is used to create a Graphic User Interface (GUI) in order for the user to not have to change the original code and be able to run different scenarios [17]. After the user enters the inputs into the GUI, the RDF is executed. The BTS OTP data is then read by the RDF and all the information needed is extracted and calculated. The scores for each day in the year are then calculated and the day with the best score is outputted by the RDF. Output files containing information the user may use are also created. The following figure provides a summary of the flow of the RDF:

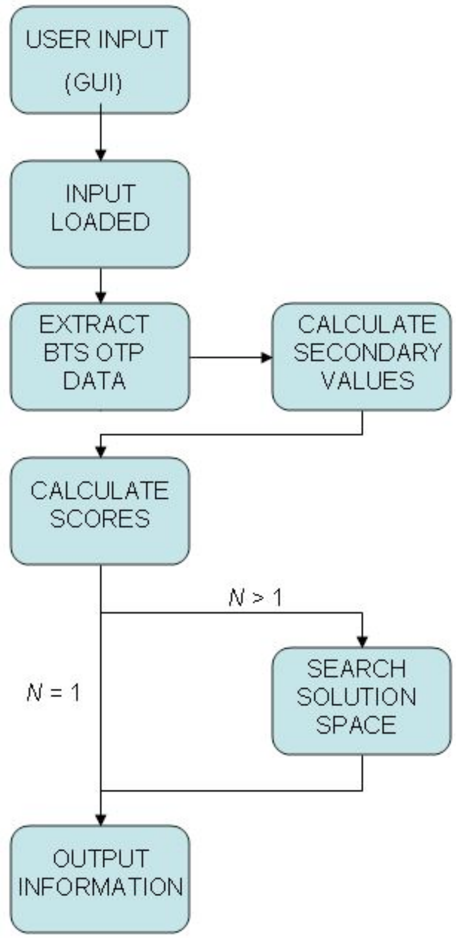

Figure 4.2: Block diagram of how the RDF works.

4.2.1 Initialization

To begin running the RDF, the RDF Python file and the BTS OTP data (using a specified naming convention) need to be in the same folder. The Python file is then executed and the following GUI will appear:

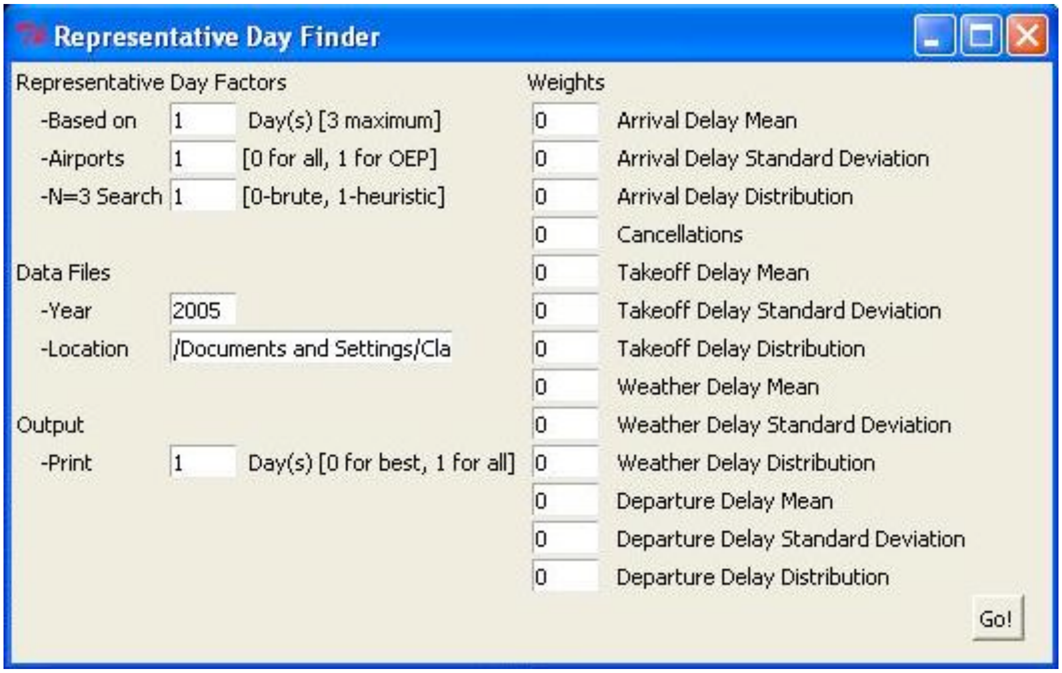

Figure 4.3: GUI that appears when the RDF is started.

The first section of inputs is labeled "Representative Day Factors." The first piece of information that needs to be inputted is the number of days of which the composite day will be composed. This means the final representative day can be based on a percentage of N days (for example, 20% Day 1, 25% Day 2, and 55% Day 3 will make the representative day for $N = 3$). The values of N that can be chosen range from '1' to '3'. More information on getting the maximum value of N to be '3' can be found in Section 6.1.1.

The next value the user can input is a '0' or '1' which will determine which airports are used in the RDF. A value of '0' means that all airports defined within the RDF will be used. A value of '1' will reduce the number of airports to 35, more specifically the OEP35. The OEP35 airports account for approximately 75% of passenger flow [14]. Most delay in the United States occurs at these 35 airports.

The following section is labeled 'Data Files' and involves the data files that are being used for data extraction. The first input in this section is the year the user wants to evaluate. This value must be consistent with the BTS OTP files in the same folder, or else the user will get an error. The next input is the folder location of the RDF file. Because files are being read and created with Python, the location of the program is called on within the RDF.

The next section is the 'Output' section. This section only contains one input for the user. The number inputted pertains to the number of days in the output file that contains the scores of the days. If the user inputs a '0', then only the best day will be outputted with its score. If the user inputs '1', then the output file will contain every day of the year with their associated scores. The user would simply have to sort the scores in order to get a grouping of days closest to representing the entire year. This is helpful in case the user needs to look at more than just a single representative day.

The final section of inputs is labeled 'Weights.' The inputs in this section are the weights that are used in the scoring function. This allows the user to put emphasis on certain factors compared to others. By inputting a '0', that factor will essentially not be considered. There are weights for all of the factors that the RDF takes into account.

Once the inputs have been entered, a button in the bottom right of the GUI will execute the RDF upon pressing.

4.2.2 Obtaining Scores

After the user inputs the necessary information, it becomes a matter of waiting on the user's part. The RDF begins by extracting the delay quantities from the BTS OTP data. After the data is extracted, the taxi-out delay is calculated from the extracted data (see Section 4.1.2). For each day and factor combination, the mean, standard deviation, and distribution are calculated. These values are also calculated for the entire year.

The next step in the RDF is to compare each day to the year. After the comparison is made, the comparison value is stored as the score factors. Once all of the score factors have been found, they are used with the user-inputted weights for the scoring function. The scoring function then assigns each day a score factor (see Section 4.1.3). After the scores have been calculated, the RDF can go down one of three paths, depending on the value of N that was inputted.

4.2.3 *N* Days

The first case of the RDF is an N value of '1.' This is the simplest case because the calculations have already been done. At this point each day has a score that represents how well it compares to the year ('well' meaning how close the day is in representing the year). Because $N = 1$ means that the best day consists of percentages of a single day, the day with the lowest score is the day that best represents the year. The day with the lowest score is found and any other days (if the user wants more than one day) are also found.

The next case is that of a value of '2' for N. A brute force method was first used to find the best combination of two days because of the nonlinearity of the score solution space:

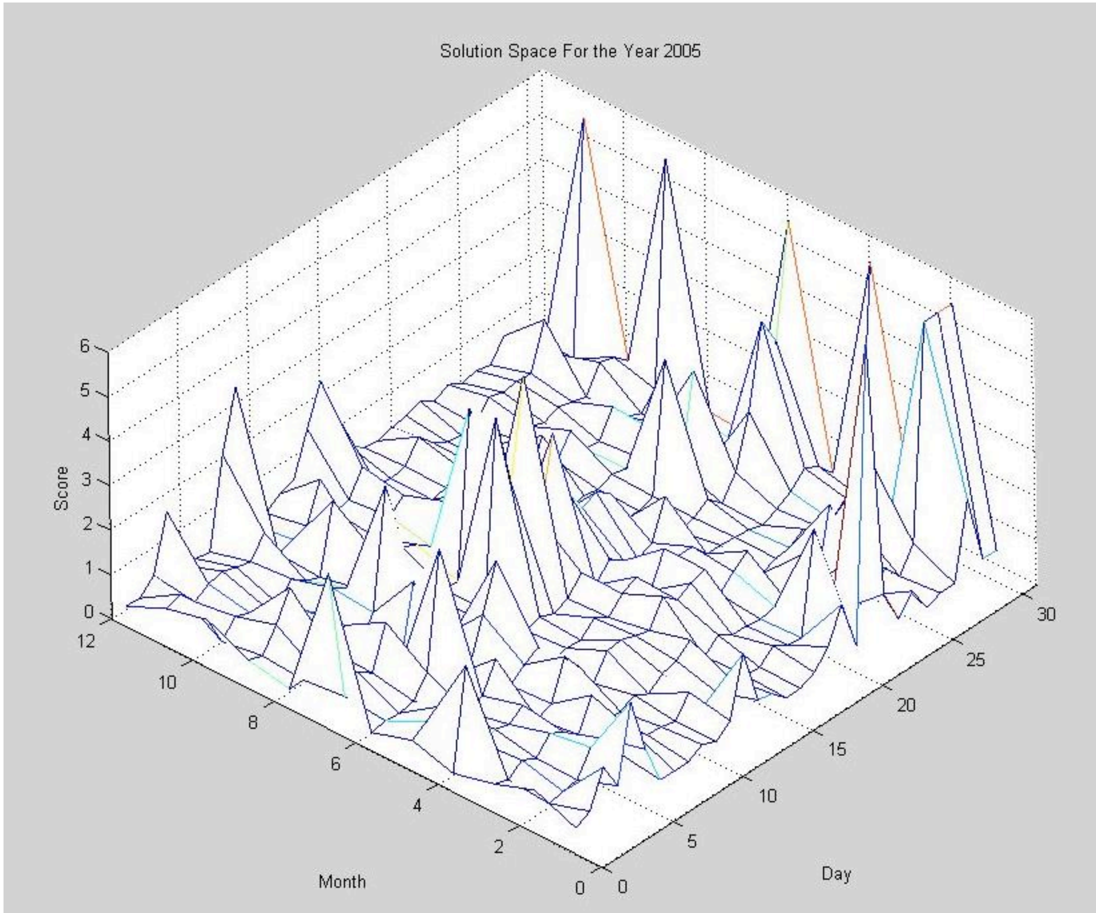

Figure 4.4: Solution space for 2005 case with all quantities weighted equally.

This meant looking at every two-day combination with percentages in intervals of 10% (starting at 10% then going to 90%). The 0% and 100% cases were ignored because of their equivalence of being $N = 1$. When the two-day combination and percentages were set, the percentages were multiplied to that day's score and summed. This gives the score of the day that was created by percentages of two days.

The last case for the RDF is when N is equal to '3.' As with the $N = 2$ case, a brute force technique was used to find the scores of all possible combinations of three days and 10%-interval percentages. Because the brute force method took too long to run (see Section 6.1.1), a heuristic search method was created to search the solution space in hopes of decreasing runtime.

Due to the complexity of the $N = 3$ solution space, a linear optimization technique cannot be used to find the global minimum. The heuristic search method that was used was a combination of particle swarm and local searches. A particle would consist of the sum of three unique days' scores (referred to as total score). The method begins by randomly generating a group of particles (combinations of days and percentages). The next step involves taking a particle and conducting a local search of the area around it. This is done by changing one of the days and recalculating the total score. If any of the new combinations have a lower total score, then the local search reiterates around the new set of days. This continues until the lowest total score in the area is found. The process is done for all of the originally generated particles. The combination with the lowest total score is the best combination the heuristic technique can find in that instance. The best combination of days from the heuristic technique is not guaranteed to be the global minimum score, but the accuracy was traded off for runtime reduction. Results from a tradeoff study can be found in Section 6.1.2.

4.2.4 Output

After the best day/combination of days is found, the RDF will output files relating to that day and the year. The delay distributions are outputted in CSV files and can be easily post-processed into graphs. For the $N = 2$ and $N = 3$ cases, the distributions for all N days are outputted. In addition, derived information from the BTS OTP data is outputted. The base taxi out time for every airport for each month is outputted in CSV files as well. Another output file contains the taxi out delay, number of flights, and average taxi out delay per flight at each airport for every day of the year. The derived data is not vital information, but can be used if needed by the user.

Finally, the representative day is outputted in a CSV file with its score that was found from the score function. If the user requests more days, this file will contain a list of these days. For the $N = 2$ and $N = 3$ cases, the days and their associated percentages will be outputted. Only one combination of days is outputted because finding the second best combination (and any more that follow) would require more computing time.

CHAPTER 5

MEANS SCENARIO AND OUTPUT DATA

The simulation tool MEANS was run to check its ability to model the historical data found in the BTS OTP files. The output data from the MEANS runs was then post processed and used with the RDF output information. The final results of the comparison of the RDF and MEANS output can be found in Chapter 6.

5.1 MEANS SCENARIO

To begin, the year 2005 was chosen for the MEANS scenario because it was the most recent year in which complete BTS OTP data was available. Using support code found within the MEANS package, weather and schedule input files were created for each day in 2005. The code extracted data from National Oceanic and Atmospheric Administration (NOAA) weather data and BTS OTP data to build input weather and schedule files, respectively, for MEANS [15].

Once the input files were generated, the simulation itself was able to be run. MEANS comes with a set of data files (such as airport and aircraft information) that it uses for its simulations. These data files, along with the weather and schedule input files, were then used for the MEANS simulation of each day. The total run time for all of the days was about 17.5 hours.

5.2 OUTPUT DATA

Having obtained the output files from MEANS, the arrival and departure delay needed to be extracted. MEANS output files are in the format of OUT files, so they were converted to CSV files. After the conversion, a script was written that calculated the arrival and departure delay from the time flights spent in the arrival and departure queues, respectively. This script generated the arrival and departure delay distributions for each day as well as the delay distributions for the entire year. With the delay distributions for every day and the entire year, a comparison between a day and the year is now possible. After running the RDF, a representative day for the year will be given. This result can be used to compare that day to the year using the MEANS output data. The results of the comparison can be found in Chapter 6.

CHAPTER 6

RESULTS

The following chapter contains the results of this thesis. The first set of results was used in determining the functions of the RDF. The RDF was finalized after these results were generated during the creation of the RDF. The second set of results show the effectiveness of the RDF. A scenario was created and the RDF was used with MEANS to find how MEANS output compares to historical data.

6.1 FINALIZING THE RDF

During the creation of the RDF, the user's freedom to choose a range of input variables was kept in mind. Some of the input options had a limited range, so problems did not arise. Other options, however, needed to be tested in order to find out how much freedom the user could have for input.

6.1.1 Maximum Value for *N*

From the beginning of this thesis, the range of the value of N was known to affect the overall runtime of the RDF. While wanting the user to be able to input a large value of N, it was apparent that there was going to be a limit on the size of N. To determine the maximum value of N, the minimum value of N was chosen as the starting point and N was incremented by '1.'

The case of $N = 0$ can be thrown out because it is trivial. The case of $N = 1$ is the base of the RDF, meaning it is calculated no matter the value of N. An increase in runtime begins at the case of $N = 2$. The runtime of the brute force technique was about one minute. With such a small runtime, there is not a strong need to implement a heuristic search method.

Once again incrementing N, the case of $N = 3$ arises. The runtime for the $N = 3$ brute force method was 17.7 hours. The following table shows the tradeoff between N and runtime:

Table 6.1: Runtime associated with the runtime of *N*.

N	Additional Time [s]
1	0
2	60
3	63847

A very large increase in runtime when going from $N = 2$ to $N = 3$ using the brute force method. This makes sense due to the combinations of days and percentages that are calculated. The possible permutations for two unique days are 365 x 364, while it is 365 x 364 x 363 for three unique days [10]. A small script was written to find the possible combinations of percentages (starting from 10% going to 90%, by 10% intervals). There are 9 combinations for $N = 2$ and 32 combinations for $N = 3$. Using this information, the number of iterations the brute force method makes can be found:

Table 6.2: Breakdown of brute force iterations for *N*.

N	Combinations of Days	Combinations of Percentages	Number of Iterations
2	132860	9	1195740
3	48228180	32	1543301760

The $N = 2$ case took about one minute to run, so 1,195,740 iterations require about one minute to complete can be assumed. Using this information, the 1,543,301,760 iterations for $N = 3$ is approximately 21.5 hours. This, while not the exact time, is within the same magnitude of the RDF's time to calculate the $N = 3$ case. Therefore, the results of the tradeoff study are reasonable. Due to this large change in runtime, the $N = 4$ case was not run.

The large runtime of the $N = 3$ case is what influenced the decision to make the value of N range from '1' to '3.' Had an $N = 4$ case been run, the runtime would have been exponentially larger compared to the $N = 3$ case (a runtime of about 2 years for $N =$

4 was calculated). The runtime for the $N = 3$ case is too large for the user's convenience, so the maximum value of N was set and work began on reducing the runtime of $N = 3$.

6.1.2 Heuristic Search Technique

The brute force method originally applied to the $N = 3$ case to find the best composite day took almost 18 hours to run. A heuristic search method (see Section 4.2.3) was created to see if the runtime could be reduced and still find the same solution. The heuristic technique required a number of initial particles to be chosen, so different amounts of initial particles were used. The following figure shows the dependence of runtime with amount of initial particles (tabular data in Appendix A.1):

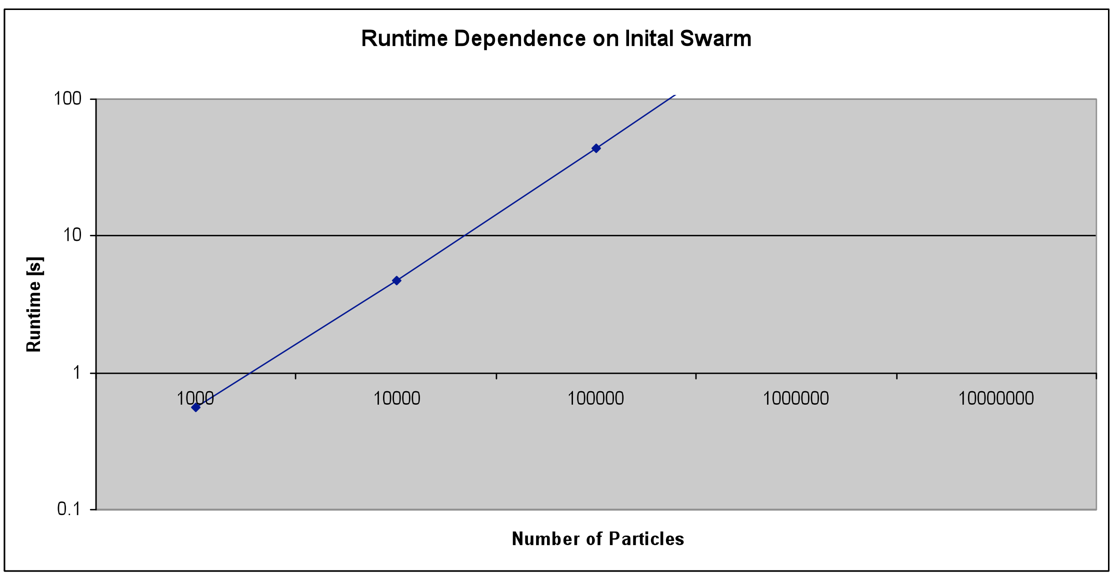

Figure 6.1: Effect of increasing initial particles on runtime.

As the number of particles was increased, the solution found by the heuristic technique moved toward the solution of the brute force method. The number of particles stopped at 10,000,000; the next increase in magnitude of 10 would have resulted in a runtime of about 24 hours. The percent difference decreased by 28% with an increase of 74.4

minutes in runtime. The results of this tradeoff study are as follows (tabular data in Appendix A.1):

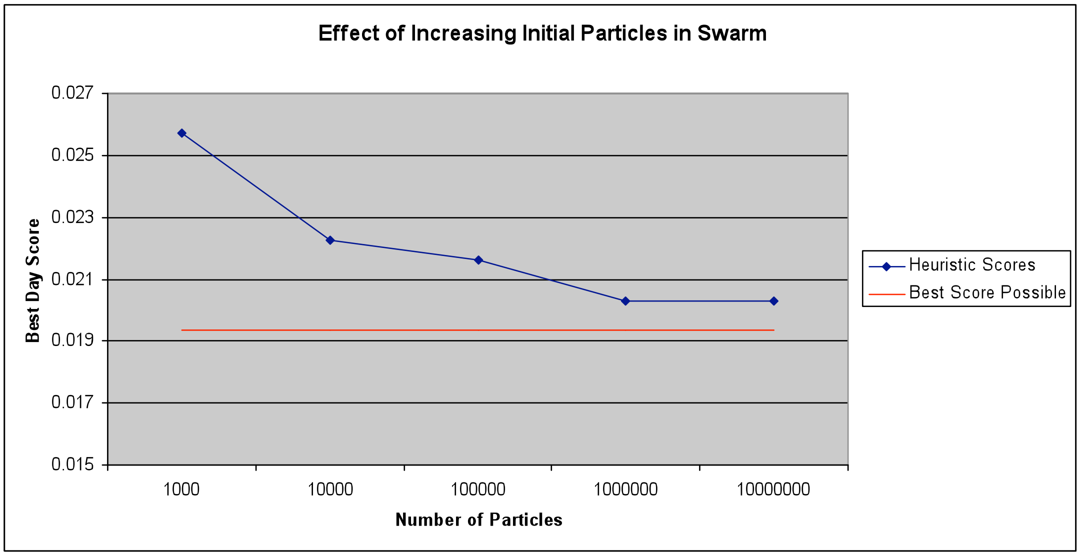

Figure 6.2: Relationship between number of initial swarm particles and the score obtained.

The best solution was still different from the brute force solution by 5%. The decision was made to implement both the brute force method as well as the heuristic search method. The input GUI of the RDF was changed so that the user can choose which method is used within the program. The user will be able to make the decision of whether runtime is more important that accuracy.

6.2 VALIDATION WITH MEANS

After the RDF was completed, the next step to take was to compare its results to those of MEANS. MEANS was run for the entire year of 2005 and the output data was collected. That data was then post-processed by using statistical analyses. Each day was given a score that determined how close it compared to the entire year. From those scores, the best representative day was found. Two scenarios were created to check the accuracy of MEANS.

The first scenario only looked at the arrival delay distribution (no other factors included). The year was 2005 and only the OEP35 airports were taken into consideration. The post-processing of the MEANS output data found the representative day to be January 9th, 2005. The RDF was then run for the same scenario and the representative day that was found was June 11th, 2005. Every day of the year was given a rank based off the MEANS scenario and June 11th was ranked 182nd (January 9th was ranked 1st, as it was the best):

Table 6.3: MEANS output data ranking comparison for arrival delay distribution.

Day	Score	Rank
January 9th	0.00014	1
May 22nd	0.000145	2
January 19th	0.00018	3
...	...	...
June 11th	0.001054	182

The comparison of the arrival distributions can be seen in the following figure:

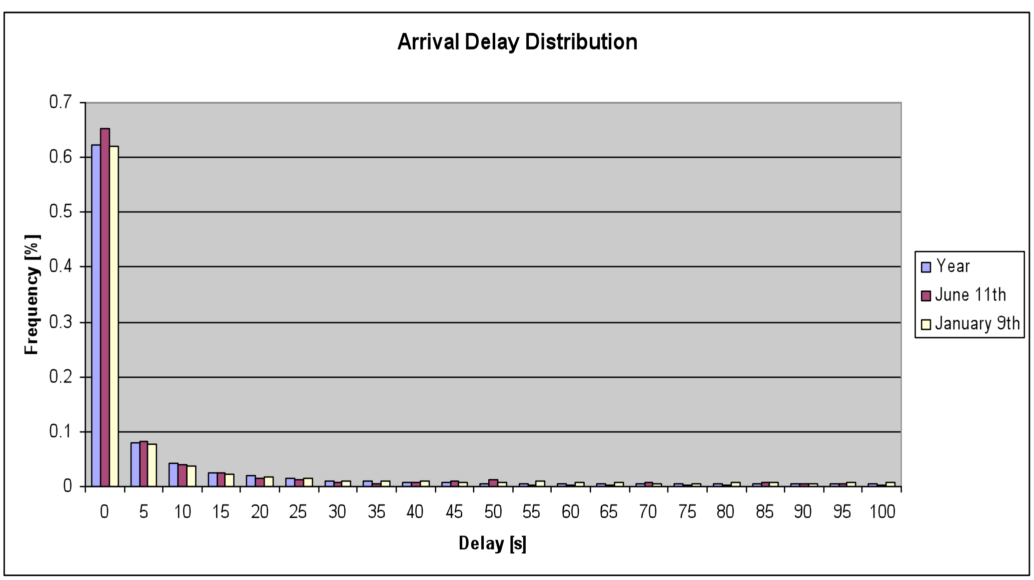

Figure 6.3: Results from RDF and MEANS output data for arrival delay.

The second scenario took into account the only the departure delay distribution. Again, the year was 2005 and the airports used were the OEP35. From the MEANS output data, the representative day was found to be June 6^{th}, 2005. After the RDF was ran using this scenario, the representative day chosen was December 2^{nd}, 2005. Using the MEANS-based rankings again, December 2^{nd} was 50^{th}.

Table 6.4: MEANS output data ranking comparison for arrival delay distribution.

Day	Score	Rank
June 6th	0.000132	1
May 2nd	0.000156	2
May 5th	0.000183	3
...	...	...
December 2nd	0.000317	50

The comparison of the distributions can be seen as follows:

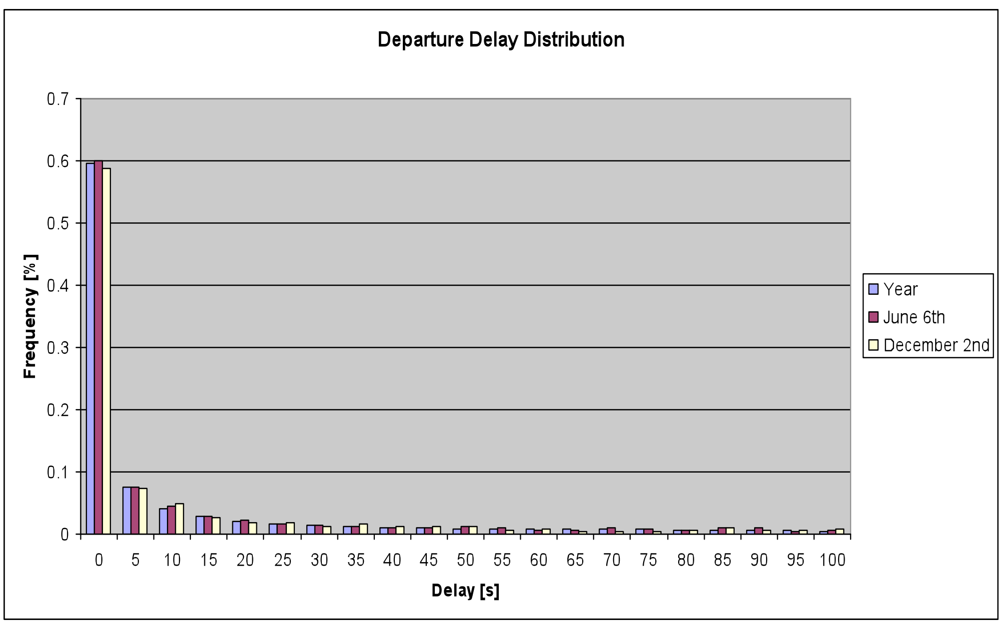

Figure 6.4: Results from RDF and MEANS output data for departure delay.

The third scenario took into account the arrival and departure delay distributions, putting equal weight on both quantities. Once again, the year was 2005 and the airports used were the OEP35. From the MEANS output data, the representative day was found to be February 8th, 2005. The RDF was run using this scenario and the day that it outputted was April 11th, 2005. Giving a ranking to the days based off of the MEANS data, April 11th was ranked 141st.

Table 6.5: MEANS output data ranking comparison for arrival and departure delay distribution.

Day	Score	Rank
February 8th	0.000561	1
April 26th	0.000566	2
August 8th	0.000571	3
...	...	...
April 11th	0.001505	141

A comparison of the distributions can be seen as follows:

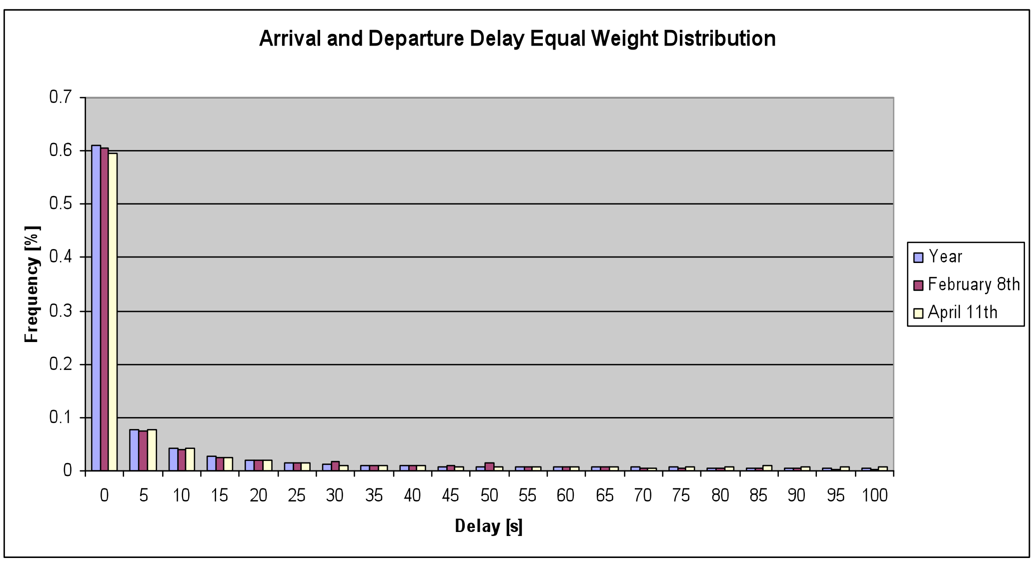

Figure 6.5: Results from RDF and MEANS output data for arrival and departure delay.

The results for checking the accuracy of MEANS met expectations. Of the three cases that were tested against the MEANS output, the best case came within the top 15%

of days that matched the year. The other two cases fell on the lower end of the top 50% of days. For the case of arrival and departure delay distribution, Appendix A.3 contains all rankings for every day of the year. These cases showed that MEANS does not perfectly model the NAS, however, that fact was already known. MEANS gives up some accuracy for low runtime, and the results show that.

Finally, a comparison of the scores from different cases of *N* was done. For the case of only looking at arrive delay distributions, the scores were found for all values of *N* (tabular data in Appendix A):

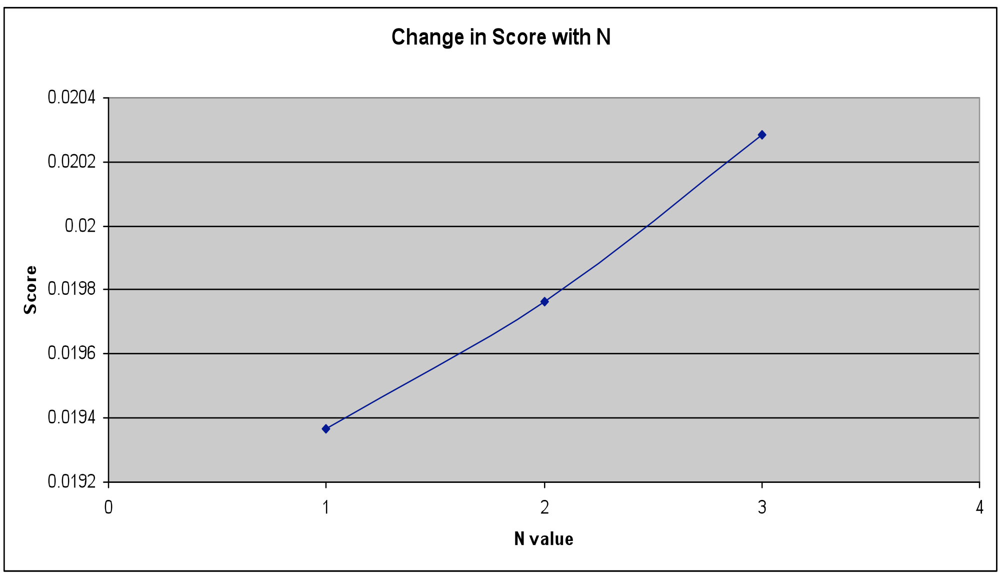

Figure 6.6: The effects of composite days on score (for one specific scenario).

The increasing values of *N* show that, for this case, a composite day does not improve the score. This does not mean that composite days will not improve the overall score (this was one scenario compared to the many other possible combinations).

CHAPTER 7

FUTURE WORK AND CONCLUSIONS

The goal of this thesis was to provide a tool capable of finding a single day that could represent an entire year for simulation purposes. This goal was met with the creation of the Representative Day Finder. While the RDF finds a day based off of historical data, there was hope that the day chosen would work for models of the NAS.

7.1 FUTURE WORK

Versatility was an important factor that was taken into account when designing the RDF. Allowing for more options in a program means the program is able to satisfy the needs of different types of users. With designing the RDF, many options arouse that were not able to be incorporated. The main option that would help make the RDF of more use would be the ability to extract data from more than just BTS OTP data. Unfortunately data sources do not have a standard convention to follow. To account for more data sources, the RDF computer code would have to be written to make specific exceptions for the data sources. This would mean locating the different data sources, extracting their data, and then converting it into the form that is used by the RDF. Doing this would require more manpower if finishing in a reasonable time is desired.

Another aspect of the RDF that could be improved is the search technique used for searching the solution space for a composite day made of three days. Many heuristics exist and may be more powerful than the one used in this thesis. The criteria for a new heuristic would be keeping runtime low and consistently finding a better solution. To do this would greatly improve the effectiveness of the RDF.

The RDF can also be used to find the accuracy of other simulation models. This thesis only used MEANS to compare with the RDF. Other simulation models exist and using their data would be a great way to test the RDF's effectiveness. The help of someone with knowledge about running other models would help in getting output data that could be post-processed and compared to the RDF results.

Lastly, a general decrease in runtime would improve upon the design of the RDF. One way to reduce runtime would be to use better programming techniques. By using effective programming, the runtime of a program can be reduced. Runtime can also be reduced by implementing more efficient search techniques, as mentioned before. Reducing the runtime of the RDF is critical because the shorter the runtime is, the more times the RDF can be run (if needed).

7.2 CONCLUSIONS

The RDF provides a foundation for work that is important to the modeling aspect of the aerospace industry. When projects that involve modeling the NAS are started, one of the first action items is to figure out a baseline year. After the baseline is established, cases are created to test with the simulation. Certain criteria must be considered when choosing these cases. With the help of the RDF, choosing which days to run in a simulation becomes easier and time doing statistical analyses for each specific project will be reduced. The RDF provides insight into how to create programs that can find days based on certain criteria.

The finalized version of the RDF is still a program capable of giving a single day that represents a year. It provides a fair amount of versatility to the user in how that day is found. The RDF also comes with a GUI that greatly reduces the amount of work the user needs to operate the program. About 2.5 hours is required for the RDF to run, which is a reasonable amount of time to extract and analyze 2.2 gigabytes of BTS OTP data (total size of the BTS OTP files for 2005).

All in all, the RDF is an easy-to-use program that will provide a single day that represents an entire year. This leads to a reduction in runtime for projects, which is a great benefit. Improvement in efficiency for the current version of the RDF will only help in making the RDF a better tool. The Representative Day Finder will provide help, as well as benefits, to those who need it when working on projects involving NAS simulations.

APPENDIX A

TABULAR VALUES FOR GRAPHS

This section provides tabular data that can be found in graphical form in this report.

A.1 HEURISTIC SEARCH TECHNIQUE RESULTS

Table A.1: Effects of increasing the number of initial swarm particles.

Number of Days	Runtime [s]	Score	Percent Difference from Best Score [%]
1000	0.563	0.025705	33
10000	4.704	0.022241	15
100000	43.75	0.021599	12
1000000	423.062	0.020284	5
10000000	4463.235	0.020284	5

A.2 CHANGES IN SCORE WITH *N*

Table A.2: Change in score with increasing *N* (for arrival delay distribution only).

N	Score
1	0.019364
2	0.019764
3	0.020284

A.3 MEANS AND RDF RANKINGS COMPARISON

Table A.3: Ranking of every day of the year for MEANS and RDF.

RDF			MEANS		
Month	Day	Rank	Month	Day	Rank
4	11	1	2	8	1
1	16	2	4	26	2
9	25	3	8	21	3
7	12	4	5	22	4
4	23	5	5	8	5
10	16	6	10	13	6
10	9	7	2	20	7
6	7	8	4	19	8

6	8	9	10	7	9
7	10	10	11	14	10
8	17	11	2	9	11
3	2	12	5	15	12
1	11	13	7	17	13
8	2	14	11	15	14
8	10	15	9	5	15
9	19	16	2	15	16
7	24	17	3	23	17
6	13	18	3	15	18
11	9	19	2	22	19
5	25	20	2	23	20
8	20	21	10	3	21
6	20	22	2	2	22
6	14	23	6	26	23
6	22	24	1	18	24
8	3	25	6	5	25
8	21	26	1	16	26
6	1	27	5	1	27
2	16	28	4	12	28
3	8	29	9	19	29
4	24	30	12	28	30
2	23	31	3	31	31
1	10	32	10	21	32
3	22	33	2	1	33
3	21	34	7	4	34
5	11	35	1	31	35
1	27	36	8	28	36
8	25	37	5	30	37
2	19	38	6	19	38
3	29	39	8	7	39
11	30	40	4	7	40
6	23	41	11	23	41
2	3	42	9	23	42
2	10	43	1	17	43
4	8	44	10	17	44
1	21	45	7	31	45
7	23	46	10	5	46
3	12	47	2	14	47
12	10	48	3	17	48
7	9	49	12	1	49
11	13	50	3	16	50
9	26	51	4	25	51
9	29	52	5	31	52
5	26	53	1	14	53
12	7	54	11	7	54
10	13	55	12	15	55
6	16	56	10	6	56

8	22	57	2	28	57
2	27	58	11	18	58
12	2	59	4	1	59
7	6	60	5	3	60
6	17	61	2	24	61
3	10	62	11	21	62
6	24	63	2	16	63
8	23	64	3	22	64
8	6	65	6	12	65
1	28	66	3	18	66
11	21	67	1	19	67
10	23	68	3	2	68
1	26	69	4	24	69
6	19	70	2	17	70
2	9	71	8	14	71
6	2	72	2	3	72
11	6	73	4	28	73
3	19	74	11	17	74
10	2	75	4	4	75
2	4	76	2	27	76
3	27	77	3	3	77
7	29	78	4	29	78
2	13	79	9	1	79
3	7	80	4	17	80
10	24	81	4	21	81
11	18	82	4	15	82
12	6	83	4	3	83
3	20	84	3	30	84
3	4	85	3	27	85
6	3	86	10	4	86
7	11	87	11	16	87
9	2	88	12	5	88
8	12	89	11	3	89
9	14	90	9	29	90
11	17	91	5	17	91
1	18	92	3	1	92
7	30	93	10	25	93
1	9	94	3	4	94
2	17	95	8	24	95
5	15	96	10	28	96
1	31	97	2	6	97
6	21	98	9	9	98
7	22	99	10	11	99
10	6	100	12	23	100
10	25	101	4	13	101
3	16	102	3	25	102
7	28	103	10	20	103
7	8	104	2	18	104

8	26	105	5	10	105
2	2	106	7	24	106
2	15	107	3	20	107
4	7	108	11	28	108
2	25	109	11	4	109
2	7	110	3	28	110
8	29	111	3	7	111
11	20	112	9	25	112
2	14	113	7	10	113
12	11	114	4	20	114
7	7	115	3	14	115
3	13	116	8	30	116
3	24	117	6	7	117
5	5	118	5	2	118
3	9	119	11	2	119
10	17	120	4	5	120
6	4	121	9	12	121
4	30	122	3	9	122
8	1	123	10	31	123
6	5	124	11	11	124
6	11	125	5	5	125
6	12	126	4	8	126
1	12	127	11	9	127
6	26	128	2	11	128
5	13	129	4	14	129
10	8	130	1	11	130
12	12	131	3	8	131
11	23	132	4	6	132
11	10	133	5	24	133
9	23	134	3	13	134
7	31	135	9	14	135
3	18	136	4	18	136
8	11	137	9	22	137
3	31	138	3	21	138
10	12	139	3	6	139
11	14	140	2	10	140
6	15	141	4	11	141
10	10	142	2	13	142
5	12	143	5	13	143
4	28	144	3	26	144
10	21	145	9	28	145
3	1	146	9	21	146
10	28	147	2	25	147
1	7	148	3	24	148
1	17	149	10	27	149
1	24	150	9	7	150
10	14	151	10	10	151
4	1	152	1	13	152

2	20	153	9	26	153
11	11	154	7	26	154
12	14	155	10	19	155
10	5	156	5	16	156
5	19	157	9	20	157
6	25	158	10	24	158
7	26	159	12	27	159
3	26	160	8	26	160
12	3	161	1	27	161
8	18	162	10	12	162
5	23	163	11	29	163
5	22	164	12	22	164
10	11	165	5	12	165
5	31	166	9	16	166
1	20	167	12	29	167
2	11	168	6	18	168
1	15	169	1	23	169
5	1	170	1	9	170
12	25	171	9	27	171
8	19	172	8	20	172
12	1	173	7	16	173
12	24	174	5	27	174
8	13	175	9	2	175
1	23	176	11	22	176
3	14	177	9	15	177
1	1	178	8	18	178
7	20	179	11	10	179
5	24	180	7	20	180
8	7	181	3	29	181
6	6	182	9	11	182
1	19	183	5	6	183
4	22	184	1	20	184
5	27	185	1	25	185
11	7	186	9	18	186
7	19	187	12	30	187
9	6	188	10	14	188
3	6	189	9	30	189
9	1	190	6	24	190
7	5	191	11	27	191
9	9	192	10	9	192
2	26	193	8	23	193
1	29	194	8	25	194
8	24	195	4	27	195
10	27	196	6	23	196
1	8	197	1	12	197
4	4	198	12	16	198
7	2	199	8	29	199
4	25	200	8	9	200

10	20	201	12	26	201
11	26	202	12	14	202
2	22	203	8	31	203
5	8	204	7	12	204
9	28	205	1	2	205
7	27	206	8	13	206
9	15	207	3	10	207
11	3	208	6	8	208
8	30	209	1	8	209
11	15	210	7	27	210
12	31	211	1	24	211
6	9	212	7	22	212
8	14	213	6	2	213
8	31	214	7	25	214
1	30	215	4	10	215
6	18	216	6	21	216
3	25	217	2	12	217
4	27	218	6	20	218
9	22	219	12	2	219
9	20	220	12	8	220
9	16	221	10	18	221
4	15	222	6	30	222
11	16	223	3	11	223
5	18	224	11	30	224
11	4	225	6	1	225
11	1	226	6	13	226
10	22	227	6	25	227
9	21	228	12	12	228
10	31	229	5	25	229
11	22	230	7	7	230
10	18	231	12	19	231
9	30	232	5	19	232
3	30	233	6	28	233
11	19	234	3	12	234
5	20	235	10	23	235
12	13	236	9	8	236
3	3	237	10	26	237
12	18	238	8	10	238
2	28	239	9	13	239
3	23	240	1	30	240
4	20	241	5	18	241
7	3	242	1	4	242
5	6	243	4	2	243
4	26	244	5	4	244
12	19	245	2	21	245
10	19	246	11	6	246
9	11	247	6	9	247
10	26	248	11	1	248

10	4	249	11	8	249
4	29	250	7	29	250
9	18	251	8	2	251
8	15	252	7	23	252
11	27	253	5	23	253
5	9	254	6	29	254
8	9	255	6	6	255
4	5	256	5	11	256
9	13	257	5	20	257
10	7	258	9	6	258
8	4	259	6	22	259
10	3	260	8	6	260
3	5	261	8	12	261
5	28	262	7	14	262
2	18	263	10	2	263
4	14	264	6	15	264
8	16	265	2	7	265
8	28	266	7	11	266
7	21	267	2	4	267
3	17	268	8	22	268
4	2	269	10	30	269
9	8	270	6	14	270
4	21	271	1	28	271
2	1	272	4	22	272
4	6	273	1	21	273
2	12	274	7	28	274
8	5	275	7	9	275
3	15	276	11	13	276
6	27	277	7	2	277
4	3	278	6	27	278
12	5	279	7	19	279
5	21	280	7	6	280
4	12	281	7	21	281
12	30	282	1	6	282
9	12	283	5	26	283
6	29	284	12	9	284
4	13	285	11	20	285
12	27	286	6	10	286
5	2	287	1	10	287
2	8	288	8	5	288
3	11	289	6	11	289
8	27	290	7	30	290
5	14	291	8	3	291
11	24	292	12	21	292
11	25	293	8	4	293
12	23	294	10	16	294
4	10	295	2	5	295
5	30	296	7	13	296

9	24	297	5	9	297
12	8	298	8	17	298
10	15	299	7	1	299
5	4	300	3	5	300
7	18	301	12	20	301
10	1	302	1	26	302
6	28	303	11	26	303
12	29	304	8	16	304
5	16	305	2	19	305
4	17	306	6	16	306
2	6	307	6	17	307
7	4	308	8	19	308
7	13	309	1	22	309
10	30	310	7	5	310
11	2	311	12	7	311
1	4	312	10	22	312
11	5	313	3	19	313
2	24	314	8	27	314
12	20	315	6	4	315
3	28	316	5	28	316
11	8	317	7	8	317
9	27	318	8	8	318
9	5	319	12	6	319
1	25	320	5	7	320
1	13	321	8	11	321
1	22	322	7	18	322
9	7	323	8	1	323
9	17	324	12	4	324
5	10	325	5	21	325
7	25	326	6	3	326
12	26	327	1	15	327
7	16	328	12	18	328
4	18	329	10	8	329
6	10	330	12	13	330
8	8	331	1	7	331
11	29	332	4	23	332
1	14	333	7	15	333
5	17	334	12	25	334
10	29	335	1	3	335
9	3	336	5	14	336
2	5	337	2	26	337
9	10	338	10	1	338
7	14	339	10	15	339
7	17	340	8	15	340
12	21	341	10	29	341
4	16	342	1	5	342
5	7	343	12	11	343
5	3	344	4	30	344

5	29	345	11	5	345
12	17	346	11	25	346
4	19	347	9	17	347
7	15	348	4	16	348
1	6	349	4	9	349
12	4	350	11	12	350
4	9	351	9	3	351
1	2	352	11	19	352
12	28	353	1	1	353
12	22	354	5	29	354
1	5	355	12	17	355
11	12	356	7	3	356
2	21	357	1	29	357
1	3	358	11	24	358
11	28	359	12	24	359
7	1	360	12	31	360
12	15	361	9	24	361
9	4	362	12	10	362
12	9	363	9	10	363
12	16	364	12	3	364
6	30	365	9	4	365

REFERENCES

[1] Couluris, George J, et al. National Airspace System Simulation Capturing the Interactions of Air Traffic Management and Flight Trajectories. AIAA Modeling and Simulation Technologies Conference and Exhibit, Austin, Texas. August 2003.

[2] Clarke, John-Paul, Terran Melconian, Elizabeth Bly, and Fabio Rabanni. 2007. MEANS-MIT Extensible Air Network Simulation. SIMULATION, 83 (5): 385-400.

[3] Hoffman, Bob, et al. A Cluster Analysis to Classify Days in the National Airspace System. AIAA Guidance, Navigation, and Control Conference and Exhibit, Austin, Texas. August 2003.

[4] Hutchison, David W, and Stacy D. Hill. Simulation Optimization of Airline Delay with Constraints. Proceedings of the 2001 Winter Simulation Conference. 2001.

[5] Kleinman, Nathan L, Stacy D. Hill, and Victor A. Ilenda. SPSA/SIMMOD Optimization of Air Traffic Delay Cost. Proceedings of the American Control Conference, Albuquerque, New Mexico. June 1997.

[6] Bazargan, Massoud, Kenneth Flemming, and Prakash Subramanian. A Simulation Study to Investigate Runway Capacity Using TAAM. Proceedings of the 2002 Winter Simulation Conference. 2002.

[7] Post, Joseph, et al. The Modernized National Airspace System Performance Analysis Capability (NASPAC). The 26th Congress of International Council of the Aeronautical Sciences (ICAS), Anchorage, Alaska. September 2008.

[8] Frolow, Igor, and Joseph H. Sinnott. National Airspace System Demand and Capacity Modeling. IEEE Log Number 8931008. May 1989.

[9] RITA | Bureau of Transportation Statistics (BTS). 2 Feb. 2009 <http://www.bts.gov/>.

[10] Walpole, R., Myers, R., Myers, S., and Ye, K., Probability & Statistics for Engineers &Scientists, 8th ed., Pearson Prentice Hall, New Jersey, 2007, Chaps. 5, 6.

[11] Idris, Husni, John-Paul Clarke, Rani Bhuva, and Laura Kang. 2002. Queuing Model for Taxi-Out Time Estimation. Air Traffic Control Quarterly 10 (1): 122.

[12] Bertsimas, Dimitris, and Tsitsiklis, John N. Introduction to Linear Optimization. Athena Scientific, Belmont, Massachusetts, 1997.

[13] Vanderplaats, Garret N. Multidiscipline Design Optimization. Vanderplaats Research & Development, Inc., Colorado Springs, CO, 2007.

[14] OEP 35 Airports. 22 Oct. 2009 <http://www.faa.gov/about/office_org/headquarters_offices/ato/publications/oep/version1/solutionsets/oep35/>.

[15] NOAA – National Oceanic and Atmospheric Administration. 29 Oct. 2009 <http://www.noaa.gov>.

[16] NASA Ames Aviation Systems Division: Highlights Archive. 27 Oct. 2009 <http://www.aviationsystemsdivision.arc.nasa.gov/news/highlights/af_highlights_20071113.shtml>.

[17] Shipman, John W. Tkinter 8.4 reference: a GUI for Python. 20 Jul. 2009 <http://infohost.nmt.edu/tcc/help/pubs/tkinter/>.

[18] "Modeling of Vehicle Environmental Characterisitcs Including New Technologies/Concepts for the Next Generation Air Transportation System." Solicited Proposal in Response to NRA NNH06ZEA001N-SSFW2 Topic A.2.7. 11 May 2007.